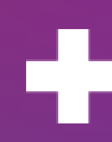
МИГРАЦИИ МЕЖДУ СХЕМАМИ И ОБНОВЛЕНИЯ В MYSQL

МЕХАНИЗМЫ, БЛОКИРОВКИ, НАГРУЗКИ

НИКОЛАЙ ИХАЛАЙНЕН

PERCONA





О ЧЁМ ЭТОТ ДОКЛАД

- зачем менять структуру
- schema / schemaless
- Online DDL
- pt-online-schema-change
- gh-ost
- Репликация и кластеры



ЗАЧЕМ МЕНЯТЬ СТРУКТУРУ БАЗЫ ДАННЫХ

- CREATE INDEX: делать запросы быстрее
- ADD FOREIGN KEY/CHECK: больше целостности, делать запросы медленнее
- Новая версия приложения:
  - ADD COLUMN
  - CHANGE COLUMN
  - RENAME COLUMN/INDEX
- CONVERT/ … CHARACTER SET …
  - миграция на UTF8/UTF8MB4



ЗАЧЕМ? УДАЛИТЬ НЕНУЖНОЕ

- FORCE/ENGINE=InnoDB: дефрагментация
- DROP INDEX, DROP COLUMN: удалить ненужное
- ADD/DROP/REORGANIZE/EXCHANGE PARTITIONS: ротация данных



ЗАЧЕМ? ОРГАНИЗАЦИЯ ДАННЫХ

- ENCRYPTION='Y'
- TABLESPACE
- COMPRESSION



СЛОЖНОСТЬ

- Изменение схемы — другой формат хранения данных на диске
  - Каждую строчку надо поменять
  - Иногда в нескольких таблицах (FOREIGN KEYS)
- Большое количество изменений



РЕЛЯЦИОННАЯ СХЕМА

- Каждая колонка
  - один тип данных (INT, BIGINT, VARCHAR, BINARY, ...)
  - дублируется в других таблицах
- Изменение класса в приложении
  - заставляет менять много таблиц в БД



ДОКУМЕНТЫ

- JSON/BSON
- Меньше "таблиц"
  - документы содержат массивы
- Приложение должно работать со старыми и новыми документами
- Индексы всё равно нужны
  - Надо делать новые/удалять при изменении структуры документа



ДОКУМЕНТЫ С МАССИВАМИ

```
CREATE TABLE customers (
    id BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    modified DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    custinfo JSON
    );

ALTER TABLE customers ADD INDEX zips( (CAST(custinfo->'$.zipcode' AS UNSIGNED ARRAY)) );
-- {"user": "Bob", "user_id": 31, "zipcode": [94477, 94536]}
SELECT id FROM customers WHERE 94568 MEMBER OF(custinfo->'$.zipcode');
-- Индекс не используется
SELECT id FROM customers WHERE 94568 MEMBER OF(custinfo->'$.zipcode[*]');
```



ВАЛИДАЦИЯ JSON-ДОКУМЕНТОВ

```
CREATE TABLE geo (
    coordinate JSON,
    CHECK(
        JSON_SCHEMA_VALID(
            '{
                "type":"object",
                "properties":{
                      "latitude":{"type":"number", "minimum":-90, "maximum":90},
                      "longitude":{"type":"number", "minimum":-180, "maximum":180}
                },
                "required": ["latitude", "longitude"]
            }',
            coordinate
        )
    )
```



MYSQL X

- X DevAPI
- X Protocol
- X Plugin



MYSQL - ДОКУМЕНТООРИЕНТИРОВАННАЯ БД

```
Type '\help' or '\?' for help; '\quit' to exit.
 MySQL  JS > var test_conn = require('mysqlx');
 MySQL  JS > var session = mysqlx.getSession({host: 'localhost', user: 'root', password: 'secret', port: 33060});
 MySQL  JS > test_collection = session.getSchema('test').createCollection("people");
<Collection:people>
 MySQL  JS > test_collection.add({birth:"1988-06-12", Name: "Francisco"});
Query OK, 1 item affected (0.2049 sec)
 MySQL  JS > test_collection.add({birth:"2001-11-03", Name: "Maria", Nickname: "Mary"});
Query OK, 1 item affected (0.1156 sec)
```



СЛОЖНОСТИ БЕЗ СХЕМЫ

- Только функциональные индексы
  - Легко ошибиться в именах полей
- Данные неожиданного типа
  - ждём целое число, в данных hex-строка
  - помогает: Валидация
- Денормализованные данные:
  - где правильное поле?
  - "распухание" размера БД
  - помогает: думать документами, а не объектами



ОБНОВЛЕНИЕ ПРИЛОЖЕНИЯ

- Приложение должно
    - работать с документами всех возможных версий
    - обновлять версию документа
        - Переход на другой тип данных
        - Удалять/добавлять поля
- Нельзя удалить старый индекс до конца миграции
- Новые индексы надо добавлять



БЕЗ СХЕМЫ: ЛЕГКО

- добавлять новые поля



МИГРАЦИИ МЕЖДУ СХЕМАМИ В РЕЛЯЦИОННОЙ БАЗЕ

- Найти разницу между боевой и тестовой базой
- Составить запросы `ALTER TABLE`
- "Выключить базу"
    - или снизить нагрузку
- Накатить изменения
- Обновить приложение



СЛОЖНОСТЬ: РЕЛЯЦИОННАЯ БД

- Физическая структура строки близка к структуре таблицы
- Блокировки на уровне
    - всей базы данных
    - таблицы
    - строки
- Параллельная работа разных запросов



METADATA LOCKS

- Активные транзакции, которые
    - видели старую таблицу
        - читали через SELECT
        - писали через INSERT/UPDATE/DELETE
    - SHARED LOCK
- ALTER TABLE:
    - EXCLUSIVE LOCK



СРАВНЕНИЕ СХЕМ: MYSQLDIFF

- https://metacpan.org/pod/distribution/MySQL-Diff/bin/mysqldiff
- работает
  - даже с generated columns
- проблема с количеством цифр в bigint:

```
$ mysqldiff --host=remote.host.com --user=myaccount db1 db2
ALTER TABLE customers CHANGE COLUMN id \
 id bigint NOT NULL AUTO_INCREMENT; \
# was bigint(20) NOT NULL AUTO_INCREMENT
```



СРАВНЕНИЕ СХЕМ: MYSQLDIFF ИЗ MYSQL-UTILITIES

- mysql-utilities больше не поддерживаются
- последняя версия требует python2 и старый mysql.connector
- в mysqlsh нет поддержки diff, плагина тоже нет
- не работает с MySQL 8.0:

```
$ mysqldiff --server1=root:secret@10.218.29.55 \
            --server2=root:secret@10.218.29.66 test:test
# WARNING: Using a password on the command line interface can be insecure.
# server1 on 10.218.29.55: ... connected.
# server2 on 10.218.29.66: ... connected.
ERROR: Query failed. 1146 (42S02): Table 'mysql.proc' doesn't exist
```



MYSQL WORKBENCH SCHEMA TRANSFER WISARD/MIGRATION WISARD

Не работают с mysql 8.0

```
Traceback (most recent call last):
  File "/usr/share/mysql-workbench/libraries/workbench/wizard_progress_page_widget.py", line 197, in thread_work
    self.func()
  File "/usr/lib64/mysql-workbench/modules/migration_object_migration.py", line 117, in task_migrate
    self.main.plan.migrate()
  File "/usr/lib64/mysql-workbench/modules/migration.py", line 510, in migrate
    self.migrationTarget.catalog =
 self.migrationSource.migration.migrateCatalog(self.state, self.migrationSource.catalog)
SystemError: AttributeError("'NoneType' object has no attribute 'name'"):
 error calling Python module function DbMySQLMigration.migrateCatalog
ERROR: Migrate Selected Objects: AttributeError("'NoneType' object has no attribute 'name'"):
 error calling Python module function DbMySQLMigration.migrateCatalog
Failed
```



РУЧНОЕ СРАВНЕНИЕ БАЗЫ ДАННЫХ

```
mysqldump --no-data \
  --set-gtid-purged=OFF --triggers --routines --events \
  --skip-add-locks --skip-lock-tables \
  --skip-lock-all-tables \
  --skip-add-drop-database --skip-add-drop-table \
  --host 10.218.29.55 test
```



РУЧНОЕ СРАВНЕНИЕ БАЗЫ ДАННЫХ

```
$ diff -up <(mysqldump ...) <( mysqldump ... )
 -- MySQL dump 10.13  Distrib 8.0.23, for Linux (x86_64)
 --
--- Host: 10.218.29.55    Database: test
+-- Host: 10.218.29.66    Database: test
--- Server version    8.0.23
+-- Server version    8.0.17

 CREATE TABLE `customers` (
-  `id` bigint NOT NULL AUTO_INCREMENT,
+  `id` bigint(20) NOT NULL AUTO_INCREMENT,
```



КАК ПРИМЕНИТЬ ИЗМЕНЕНИЯ?

- Всё сразу
- Табличка за раз
- Одно изменение за раз



ВСЁ СРАЗУ

- удалить все данные
- создать таблицы с новой структурой
- восстановить данные из резервной копии
- полезно, если:
    - меняется тип FOREIGN KEY и много дочерних таблиц
    - меняются тысячи маленьких таблиц



ТАБЛИЧКА ЗА РАЗ

- Собрать все изменения таблицы в один ALTER

```
ALTER TABLE customers \
  ADD COLUMN is_valid TINYINT AFTER modified, \
  ADD KEY (modified, is_valid);
```

- Самый эффективный способ работы



ОДНО ИЗМЕНЕНИЕ ЗА РАЗ

- ниже эффективность
- удобно для разработки
- необходимо в тестах производительности



ALTER TABLE ALGORITHM=...

- COPY
- INPLACE
- INSTANT



ALTER TABLE ALGORITHM=COPY

- заблокировать все операции над таблицей
- создать временную таблицу в директории с базой данных
- копировать данные
- поменять местами таблицы, удалить старую таблицу



PERFORMANCE SCHEMA

```
mysql> ALTER TABLE t1 FORCE, ALGORITHM=COPY;
mysql> select EVENT_ID,EVENT_NAME,SOURCE from events_stages_history_long \
       where thread_id=47 and event_id between 445573 and 18480686 \
       order by event_id asc;
+----------+-------------------------------------------------+-----------------------------------+
| EVENT_ID | EVENT_NAME                                      | SOURCE                            |
+----------+-------------------------------------------------+-----------------------------------+
|   445573 | stage/sql/starting                              | init_net_server_extension.cc:101 |
|   445577 | stage/sql/Executing hook on transaction begin.  | rpl_handler.cc:1378               |
|   445578 | stage/sql/starting                              | rpl_handler.cc:1380               |
|   445580 | stage/sql/checking permissions                  | sql_authorization.cc:2200         |
|   445581 | stage/sql/checking permissions                  | sql_authorization.cc:2200         |
|   445582 | stage/sql/init                                  | sql_table.cc:15758                |
|   445584 | stage/sql/Opening tables                        | sql_base.cc:5747                  |
```



PERFORMANCE SCHEMA

```
|   445659 | stage/sql/setup                                | sql_table.cc:16262                 |
|   445737 | stage/sql/creating table                       | sql_table.cc:8675                  |
|   445738 | stage/sql/After create                         | sql_table.cc:8748                  |
|   446666 | stage/sql/System lock                          | lock.cc:332                        |
|   446671 | stage/sql/copy to tmp table                    | sql_table.cc:17112                 |
| 18475406 | stage/sql/rename result table                  | sql_table.cc:17221                 |
| 18480686 | stage/sql/end                                  | sql_table.cc:17527                 |
+----------+------------------------------------------------+------------------------------------+
```



УРОВЕНЬ ФАЙЛОВ

```
# ls -l
total 290824
-rw-r-----. 1 mysql mysql  71303168 #sql-769_9.ibd
-rw-r-----. 1 mysql mysql 226492416 t.ibd
```



УРОВЕНЬ ФАЙЛОВ

- Нет свободного места в директории базы данных?
  - ALTER вызовет ошибку
- Выполнилось на слейве?
  - ошибки репликации



ALTER TABLE ALGORITHM=INPLACE

- заблокировать DDL и DML над таблицей
- разблокировать и записывать все изменения DML в файл
  - расположенный в `innodb_tmpdir`
  - размером не больше `innodb_online_alter_log_max_size` байт
- читать данные из таблицы из PK
- сортировать данные (делаем индекс)
- вставить отсортированные данные в таблицу (b-tree)
- сбросить все модифицированные страницы табличного пространства на диск
- применить лог к новому индексу
- получить эксклюзивную блокировку на изменение таблицы
- обновить метаданные



PERFORMANCE SCHEMA

```
mysql> ALTER TABLE t1 FORCE, ALGORITHM=INPLACE;
mysql> select EVENT_ID,EVENT_NAME,SOURCE from events_stages_history_long \
       where thread_id=47 and event_id between 1796 and 444589 \
       order by event_id asc;
+----------+------------------------------------------------------+--------------------+
| EVENT_ID | EVENT_NAME                                           | SOURCE             |
+----------+------------------------------------------------------+--------------------+
|     1796 | stage/sql/System lock                                | lock.cc:332        |
|     1802 | stage/sql/preparing for alter table                  | sql_table.cc:12944 |
|     2927 | stage/sql/altering table                             | sql_table.cc:13006 |
```



PERFORMANCE SCHEMA

```
|      2929 | stage/innodb/alter table (read PK and internal sort) | ut0stage.h:213     |
|    404887 | stage/innodb/alter table (flush)                     | ut0stage.h:425     |
|    441786 | stage/innodb/alter table (log apply table)           | ut0stage.h:425     |
|    441836 | stage/sql/committing alter table to storage engine   | sql_table.cc:13048 |
|    441837 | stage/innodb/alter table (end)                       | ut0stage.h:425     |
|    441847 | stage/innodb/alter table (log apply table)           | ut0stage.h:425     |
|    444589 | stage/sql/end                                        | sql_table.cc:13191 |
+-----------+------------------------------------------------------+--------------------+
```



INPLACE ALTER TABLE FORCE

До alter table

```
-rw-r-----. 1 mysql mysql 226492416 t.ibd
```

Во время

```
-rw-r-----. 1 mysql mysql 255852544 #sql-ib1064-3964883563.ibd
-rw-r-----. 1 mysql mysql 226492416 t.ibd
```

Размер после

```
-rw-r-----. 1 mysql mysql 255852544 t.ibd
```



INPLACE ALTER TABLE: ALTER TABLE (FLUSH)

- размер временного файла не меняется
- происходит запись модифицированных страниц на диск

```
mysql> SELECT EVENT_NAME, WORK_COMPLETED, WORK_ESTIMATED \
  FROM performance_schema.events_stages_current;
+--------------------------------------+----------------+----------------+
| EVENT_NAME                           | WORK_COMPLETED | WORK_ESTIMATED |
+--------------------------------------+----------------+----------------+
| stage/innodb/alter table (flush)     |          23800 |          26087 |
+--------------------------------------+----------------+----------------+
```



INPLACE СОЗДАНИЕ ИНДЕКСА

- временный файл не нужен
- существующая таблица растёт в размерах

```
mysql> ALTER TABLE t ADD KEY(C), ALGORITHM=INPLACE;
# До ALTER TABLE
-rw-r-----. 1 mysql mysql 226492416 t.ibd
# Во время ALTER TABLE
-rw-r-----. 1 mysql mysql 427819008 t.ibd
```



INPLACE REBUILDS TABLE

- Rebuilds Table: Yes
- Rebuilds Table: No



ALTER TABLE ALGORITHM=INSTANT

```
mysql> ALTER TABLE t ADD COLUMN c2 INT DEFAULT 0, ALGORITHM=INSTANT;
mysql> select EVENT_ID,EVENT_NAME,SOURCE from events_stages_history_long \
  where thread_id=47 and event_id between 18482231 and 18484410 order by event_id asc;
| EVENT_ID | EVENT_NAME                                         | SOURCE              |
| 18482231 | stage/sql/Opening tables                           | sql_base.cc:5747    |
| 18482306 | stage/sql/setup                                    | sql_table.cc:16262  |
| 18482384 | stage/sql/creating table                           | sql_table.cc:8675   |
| 18482385 | stage/sql/After create                             | sql_table.cc:8748   |
| 18482392 | stage/sql/System lock                              | lock.cc:332         |
| 18482398 | stage/sql/preparing for alter table                | sql_table.cc:12944  |
| 18482399 | stage/sql/altering table                           | sql_table.cc:13006  |
| 18482468 | stage/sql/committing alter table to storage engine | sql_table.cc:13048  |
| 18484410 | stage/sql/end                                      | sql_table.cc:13191  |
```



ALTER TABLE ALGORITHM=INSTANT

- ADD COLUMN
  - кроме ROW_FORMAT=COMPRESSED
  - кроме таблиц с FULLTEXT
- поменять/удалить значение колонки по умолчанию (DEFAULT ...)
- изменить ENUM или SET



MARIADB

- получили Online DDL из MySQL 5.6 в наследство
- начали реализовывать алгоритм INSTANT в 2017 (10.3)



MARIADB: INSTANT

```
MariaDB [test]> SET SESSION alter_algorithm='INSTANT';
MariaDB [performance_schema]> select EVENT_NAME,SOURCE from events_stages_history_long \
        where thread_id=16 and event_id > 157 order by event_id asc;
| EVENT_NAME                                           | SOURCE |
| stage/sql/System lock                                |        |
| stage/sql/table lock                                 |        |
| stage/sql/After create                               |        |
| stage/sql/preparing for alter table                  |        |
| stage/sql/altering table                             |        |
| stage/sql/Committing alter table to storage engine   |        |
| stage/sql/Unlocking tables                           |        |
26 rows in set (0.002 sec)
```



MARIADB: INSTANT

- отлично работает
- можно менять значение поля по умолчанию (DEFAULT)
- можно менять размеры varchar, если не меняется поле длины
  - varchar(10), длина 1 байт
  - varchar(300), длина 2 байта
- можно добавлять/удалять колонку (кроме AUTO INC)
- можно изменить `AUTO_INCREMENT`
- переименовать колонку
- менять порядок колонок
- новые значения ENUM и SET
- удалить индекс
- удалить FK



MARIADB: NOCOPY

```
MariaDB [test]> SET SESSION alter_algorithm='NOCOPY';
Query OK, 0 rows affected (0.000 sec)

MariaDB [test]> alter table t add index (c);
```



MARIADB: NOCOPY

```
MariaDB [performance_schema]> select EVENT_NAME from events_stages_history_long \
        where thread_id=16 and event_id > 30965652 order by event_id asc;
| EVENT_NAME                                           |
...
| stage/sql/altering table                             |
| stage/innodb/alter table (read PK and internal sort) |
| stage/innodb/alter table (merge sort)                |
| stage/innodb/alter table (insert)                    |
| stage/innodb/alter table (log apply index)           |
| stage/sql/Committing alter table to storage engine   |
| stage/innodb/alter table (end)                       |
| stage/sql/Unlocking tables                           |
31 rows in set (0.003 sec)
```



MARIADB: NOCOPY

- как ALGORITHM=INPLACE в MySQL, то не перестраивает табличку
- это не новый режим
    - эквивалент ALGORITHM=INPLACE Rebuilds table: No



MARIADB: INPLACE

```
MariaDB [test]> SET SESSION alter_algorithm='INPLACE';
Query OK, 0 rows affected (0.000 sec)

MariaDB [test]> alter table t engine=innodb;
Query OK, 0 rows affected (27.063 sec)
Records: 0  Duplicates: 0  Warnings: 0
```



MARIADB: INPLACE

```
MariaDB [performance_schema]> select EVENT_NAME from events_stages_history_long \
        where thread_id=16 and event_id > 31047161 order by event_id asc;
+-------------------------------------------------------+
| EVENT_NAME                                            |
+-------------------------------------------------------+
| stage/sql/System lock                                 |
| stage/sql/table lock                                  |
| stage/sql/After create                                |
| stage/sql/preparing for alter table                   |
| stage/sql/altering table                              |
| stage/innodb/alter table (read PK and internal sort)  |
```



MARIADB: INPLACE

```
| stage/innodb/alter table (merge sort)                     |
| stage/innodb/alter table (insert)                         |
| stage/innodb/alter table (merge sort)                     |
| stage/innodb/alter table (insert)                         |
| stage/innodb/alter table (merge sort)                     |
| stage/innodb/alter table (insert)                         |
| stage/innodb/alter table (log apply table)                |
| stage/sql/Committing alter table to storage engine        |
| stage/innodb/alter table (end)                            |
| stage/innodb/alter table (log apply table)                |
| stage/sql/Unlocking tables                                |
+-----------------------------------------------------------+
36 rows in set (0.003 sec)
```



MARIADB: INPLACE

- создаёт большой временный файл с новой таблицей
- ENGINE=InnoDB делает merge sort + insert для каждого вторичного индекса



MARIADB: COPY

```
MariaDB [test]> SET SESSION alter_algorithm='COPY';
Query OK, 0 rows affected (0.000 sec)

MariaDB [test]> alter table t modify `c` varchar(10) DEFAULT 'c' NOT NULL;
ERROR 1406 (22001): Data too long for column 'c' at row 1
MariaDB [test]> alter table t modify `c` varchar(1000) DEFAULT 'c' NOT NULL;
```



MARIADB: COPY

```
MariaDB [performance_schema]> select EVENT_NAME from events_stages_history_long \
        where thread_id=16 and event_id > 31191068 order by event_id asc;
| EVENT_NAME                                  |
| stage/sql/System lock                       |
| stage/sql/table lock                        |
| stage/sql/After create                      |
| stage/sql/copy to tmp table                 |
| stage/sql/Enabling keys                     |
| stage/sql/Rename result table               |
| stage/sql/Unlocking tables                  |
| stage/sql/Rename result table               |
| stage/sql/End of update loop                |
| stage/sql/Query end                         |
26 rows in set (0.003 sec)
```



MARIADB: COPY

- как и в MySQL
- создаёт новое табличное пространство (файл `*.ibd`)
- копирует данные под полной блокировкой



ПРОИЗВОДИТЕЛЬНОСТЬ

- sysbench oltp-rw 10M rows, 2.4G, 10GB Buffer pool

ALTER	Время	Время+sysbench
force, algorithm=copy	1 min 57.71	1 min 57.14
force, algorithm=inplace	1 min 10.49	1 min 50.40
add key(k), algorithm=copy	3 min 1.26	3 min 5.96
add key(k), algorithm=inplace	20.72 sec	8.0.22-13 crash
drop key k, algorithm=copy	2 min 10.81	2 min 6.89
drop key k, algorithm=inplace	0.01 sec	0.01 sec



PERCONA XTRADB CLUSTER И ONLINE DDL

```
./anydbver deploy \
  node0 pxc \
  node1 pxc galera-master:node0 \
  node2 pxc galera-master:node0 \
  node3 sysbench sysbench-mysql:node0 oltp_read_write
```



PERCONA XTRADB CLUSTER И ONLINE DDL

```
pxc2> alter table sbtest1 force;
pxc0> show processlist\G
*************************** 10. row ***************************
           Id: 16
         User: system user
         Host:
           db: sbtest
      Command: Sleep
         Time: 7
        State: altering table
         Info: alter table sbtest1 force
```



PERCONA XTRADB CLUSTER И ONLINE DDL

```
*************************** 16. row ***************************
           Id: 41
         User: root
         Host: ihanick-node3.lxd:35288
           db: sbtest
      Command: Execute
         Time: 2
        State: Waiting for table metadata lock
         Info: UPDATE sbtest1 SET k=k+1 WHERE id=49905
```



PERCONA XTRADB CLUSTER И ONLINE DDL

```
*************************** 17. row ***************************
           Id: 42
         User: root
         Host: ihanick-node3.lxd:35260
           db: sbtest
      Command: Execute
         Time: 2
        State: update
         Info: INSERT INTO sbtest6 (id, k, c, pad) VALUES (49770, 49847, '73465953637-04726898262-98349612436-49298

pxc0: mysql_stmt_execute() returned error 1412 \
(Table definition has changed, please retry transaction) \
for query 'UPDATE sbtest1 SET k=k+1 WHERE id=?'
```



PERCONA XTRADB CLUSTER И ONLINE DDL

- `wsrep_OSU_method=TOI` блокирует запросы на соседних узлах
- даже если выбрали RSU:

```
pxc2> alter table sbtest1 force, ALGORITHM=INPLACE, LOCK=NONE;
ERROR 1845 (0A000): LOCK=NONE is not supported for this operation. Try LOCK=SHARED.
```

- Без Galera:

```
mysql> alter table sbtest1 force, ALGORITHM=INPLACE, LOCK=NONE;
Query OK, 0 rows affected (1.74 sec)
```



PERCONA XTRADB CLUSTER 5.7 И ONLINE DDL

```
./anydbver deploy \
  node0 pxc:5.7 \
  node1 pxc:5.7 galera-master:node0 \
  node2 pxc:5.7 galera-master:node0 \
  node3 sysbench sysbench-mysql:node0 oltp_read_write
```



PERCONA XTRADB CLUSTER 5.7 И ONLINE DDL

```
[ 28s ] thds: 8 tps: 24.49 qps: 489.85 (r/w/o: 342.89/52.48/94.47) lat (ms,95%): 475.79 err/s: 0.00 reconn/s: 0.00
[ 30s ] thds: 8 tps: 23.00 qps: 488.51 (r/w/o: 343.00/50.50/95.00) lat (ms,95%): 707.07 err/s: 1.50 reconn/s: 0.00
======= ALTER TABLE =========
[ 32s ] thds: 8 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 34s ] thds: 8 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 36s ] thds: 8 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
```



PERCONA XTRADB CLUSTER 5.7 И ONLINE DDL

```
Sleep    |    16 | altering table                   | alter table sbtest1 force, ALGORITHM=INPLACE
Execute |    16 | wsrep: initiating
                   pre-commit for write set (950)  | COMMIT
```



PERCONA XTRADB CLUSTER 5.7 И ONLINE DDL, RSU

```
[ 8s ] thds: 8 tps: 28.50 qps: 570.01 (r/w/o: 399.01/59.00/112.00) lat (ms,95%): 434.83 err/s: 0.00 reconn/s: 0.00
[ 10s ] thds: 8 tps: 32.50 qps: 650.04 (r/w/o: 455.03/66.00/129.01) lat (ms,95%): 404.61 err/s: 0.00 reconn/s: 0.00
=== ALTER TABLE на node1, sysbench на node0 ===
[ 12s ] thds: 8 tps: 34.48 qps: 654.20 (r/w/o: 457.29/68.97/127.94) lat (ms,95%): 390.30 err/s: 0.00 reconn/s: 0.00
[ 14s ] thds: 8 tps: 27.01 qps: 574.14 (r/w/o: 403.60/58.01/112.53) lat (ms,95%): 383.33 err/s: 0.00 reconn/s: 0.00
[ 16s ] thds: 8 tps: 26.50 qps: 531.51 (r/w/o: 371.01/55.00/105.50) lat (ms,95%): 450.77 err/s: 0.00 reconn/s: 0.00
```



PERCONA XTRADB CLUSTER 5.7 И ONLINE DDL, RSU, ТОТ ЖЕ УЗЕЛ ДЛЯ SYSBENCH

```
[ 12s ] thds: 8 tps: 10.50 qps: 248.00 (r/w/o: 175.00/29.00/44.00) lat (ms,95%): 502.20 err/s: 4.00 reconn/s: 0.00
=== ALTER TABLE на node0, sysbench на node0 ===
[ 14s ] thds: 8 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 16s ] thds: 8 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
```

```
sbtest | Query    |    29 | altering table                          | alter table sbtest1 force, ALGORITHM=INPLACE
sbtest | Execute  |    29 | wsrep: waiting to replay write set (-1) | COMMIT
```



PERCONA XTRADB CLUSTER 5.7 И ONLINE DDL, RSU, ТОТ ЖЕ УЗЕЛ ДЛЯ SYSBENCH

```
mysql> set session wsrep_osu_method='RSU';
mysql> set global wsrep_desync=1;
mysql> alter table sbtest1 force, ALGORITHM=INPLACE;
mysql> set global wsrep_desync=0;
```

```
sbtest | Query   |   26 | altering table                       | alter table sbtest1 force, ALGORITHM=INPLACE
sbtest | Execute |   26 | wsrep: waiting to replay write set (-1) | COMMIT
sbtest | Execute |   25 | Waiting for table metadata lock      | UPDATE sbtest1 SET c=? WHERE id=?
```



MARIADB GALERA CLUSTER

```
./anydbver deploy \
  node0 mariadb-cluster \
  node1 mariadb-cluster galera-master:node0 \
  node2 mariadb-cluster galera-master:node0 \
  node3 sysbench sysbench-mysql:node0 oltp_read_write
```



MARIADB GALERA CLUSTER

- 10.6.0
- работает `alter table sbtest1 engine=innodb, ALGORITHM=INPLACE, LOCK=NONE;`
- TOI, разные узлы

```
Command| Time     | State                       | Info
sbtest | Sleep    |   58 | altering table       | alter table sbtest1 engine=innodb, ALGORITHM=INPLACE, LOCK=NONE
sbtest | Execute |   57 | starting             | COMMIT
sbtest | Execute |   58 | wsrep replaying trx | COMMIT
sbtest | Execute |   59 | Starting cleanup     | COMMIT
```



MARIADB GALERA CLUSTER

- RSU на **другом** узле: без блокировки
- RSU на **том же** узле: COMMIT ждёт

```
MariaDB [sbtest]> SET SESSION wsrep_osu_method='RSU';
MariaDB [sbtest]> ALTER TABLE sbtest1 ENGINE=innodb, ALGORITHM=INPLACE, LOCK=NONE;
```



INNODB CLUSTER

```
./anydbver deploy \
  node0 mysql group-replication \
  node1 mysql group-replication master:node0 \
  node2 mysql group-replication master:node0 \
  node3 mysql-router master:node0 \
  node4 sysbench sysbench-mysql:node0 oltp_read_write
```



INNODB CLUSTER

- В режиме single primary работает как отдельный узел
- Параллельная модификация в режиме Multi-Primary не поддерживается



INNODB CLUSTER

- стадия 1

```
Command | Time | State
Query   |  428 | altering table | alter table sbtest1 engine=innodb, ALGORITHM=INPLACE, LOCK=NONE
```

- стадия 2: исполнение ALTER на остальных узлах

```
Command | Time | State
Query   |  616 | waiting for handler commit       | alter table sbtest1 engine=innodb, ALGORITHM=INPLACE, LOCK=NONE
Execute |    2 | Waiting for table metadata lock | SELECT DISTINCT c FROM sbtest1 WHERE id BETWEEN 44902 AND 45001 ORDER B
Execute |    2 | Waiting for table metadata lock | UPDATE sbtest1 SET c='49380704404-54896213215-83976510294-72783551747-2
```



НЕДОСТАТКИ ONLINE DDL

- не срабатывает для INPLACE
  - при наличии уникальных ключей
  - удаления+создания записи

```
ERROR 1062 (23000): Duplicate entry '257852-107432-3959699250' for key 'uniq_key_2'
```

- плохо работает с Galera/InnoDB Cluster



КАК СДЕЛАТЬ ALTER TABLE С МИНИМУМОМ БЛОКИРОВОК?

- копировать данные интервалами в теневую таблицу
- параллельно менять и старую и новую таблицу
- поменять таблицы местами
- все операции можно реализовать на SQL



PT-ONLINE-SCHEMA-CHANGE

- perl-скрипт, который использует SQL
- часто perl-DBD-MySQL собран со старой библиотекой mysql
  - для MySQL 8 нужен пользователь с `mysql_native_password`

```
CREATE USER sbtest@'%' IDENTIFIED \
WITH mysql_native_password BY 'secret';
```

- как запустить?

```
./pt-online-schema-change --execute \
--alter 'ENGINE=InnoDB' \
D=sbtest,t=sbtest1,u=sbtest,p=secret
```



PT-ONLINE-SCHEMA-CHANGE

- нужны metadata lock при создании триггеров

```
Command | Time | State
Execute |    4 | Waiting for table metadata lock | SELECT c FROM sbtest1 WHERE id BETWEEN 49988 AND 50087
Query   |    4 | System lock                     | CREATE TRIGGER `pt_osc_sbtest_sbtest1_del` AFTER DELETE ON `sbtest`.`sb
```

- отслеживает задержки при асинхронной репликации
- может работать с FOREIGN KEYS
- не работает на репликах



GH-OST

- изменения таблицы можно взять из binary log

```
gh-ost --execute \
  --allow-on-master \
  --user=root --password=secret \
  --database sbtest --table sbtest1 \
  --alter="ENGINE=InnoDB" --chunk-size=2000
```



GH-OST

- работает без триггеров
- читает binary log в формате ROW
- копирует данные на мастере в теневую таблицу
- применяет изменения к теневой таблице



НЕДОСТАТКИ GH-OST

- не работает с шифрованным binary log
- не работает с Galera
- не работает с триггерами
- нет FOREIGN KEYS



ПРЕИМУЩЕСТВА ONLINE DDL

- с каждой мажорной версией MySQL улучшается
- минимально используется диск (1x или 2x, но не 2x+1x(binary log))
- алгоритм INSTANT — идеальный вариант
- покрывает большую часть разумных изменений базы
    - индексы
    - новые колонки



ПРЕИМУЩЕСТВА PT-ONLINE-SCHEMA-CHANGE И GH-OST

- альтернатива Online DDL
- возможность снижения скорости копирования по условию
  - большая нагрузка
  - отстают реплики
- pt-online-schema-change: работает с Galera/PXC



ВОПРОСЫ

????

